LA-UR-11-2313

*Approved for public release;
distribution is unlimited.*

Title: User-Friendly Surface Flux Tallies in MCNP

Author(s): Jeffrey A. Favorite, XCP-7

Intended for: Electronic distribution and archive

This page intentionally left blank.

Los Alamos
NATIONAL LABORATORY

research note

X-Computational Physics Division
Transport Applications Group

Group XCP-7, MS F699
Los Alamos, New Mexico 87545
505/606-0495
Fax: 505/667-3726

To/MS: Distribution
From/MS: Jeffrey A. Favorite / XCP-7, MS F663
Phone/Email: 7-7941 / fave@lanl.gov
Symbol: XCP-7–RN(U)11–01 (LA–UR–11–2313)
Date: April 18, 2011

SUBJECT: User-Friendly Surface Flux Tallies in MCNP (U)

Abstract

Monte Carlo surface flux tallies use a fixed cosine value that defines the grazing range and a fixed cosine value to score with instead of a grazing cosine. In MCNP, these cosines are hard-wired, but making them user inputs would allow users to diagnose and overcome difficulties associated with grazing in surface flux (F2) tallies. This paper describes a simple means of allowing these cosines to be user inputs for each F2 tally. A new special tally treatment, MUC, is used; it has two entries, the cosine cutoff and the substitute cosine. Nine subroutines were modified and one global variable, dimensioned 2 × the number of tallies, was added. These modifications will allow MCNP to compute analytic benchmarks for surface flux tallies.

I. Introduction

Recent work[1,2] has demonstrated that the standard method of dealing with grazing cosines in Monte Carlo surface-flux tallies is either inadequate or suboptimal in some situations. Of particular concern are one-sided fluxes associated with exterior surfaces, one-way streaming (external irradiation of objects or uncollided ray-tracing), and user-input cosine bins that compute fluxes in inward and outward directions separately.

The standard estimate of the integrated flux in the grazing range, defined as the range of surface-crossing cosines $|\mu|$ less than some cutoff $\mu_{cut} > 0$, involves scoring the weight divided by a substitute cosine $\mu_{sub} > 0$ instead of the surface-crossing cosine $|\mu|$ as is done outside the grazing range. In the production Monte Carlo codes, the cosine cutoff μ_{cut} and the substitute cosine $\mu_{sub} = \mu_{cut}/2$ (the provenance of this approximation can be found in Refs. 1 and 2 and the references therein) are hard-wired.

In Ref. 2, it was shown that users can diagnose and overcome difficulties due to grazing angles if both μ_{cut} and μ_{sub} can be modified. In this paper, we document a user interface and source code modifications that were implemented to achieve this objective quite easily in the MCNP6 initial release[3] (MCNP6_Beta1).

We defined a new special tally treatment, MUC. With this new feature, surface flux tallies are now more informative and user friendly.

The user interface for the new MUC special tally treatment is presented in Sec. II. The source code modifications are presented in Sec. III. A test input file is presented in Sec. IV.

II. User Interface

Values of μ_{cut} and μ_{sub} are input for each F2 tally using the new MUC special tally treatment, as follows.

Form: FT*n* MUC C S
n = tally number ending in 2
C = cosine cutoff defining the upper limit of the grazing range (μ_{cut})
S = substitute cosine to use for scoring in the grazing range (μ_{sub})

Default: Surface flux (F2) tallies with no MUC treatment will use the historic MCNP default values, μ_{cut} = 0.10 and $\mu_{sub} = \mu_{cut}/2$. The "jump" input character J can be used to accept defaults for C, S, or both. The default value for C is 0.10. The default value for S is half of C, whether C is input or defaulted.

The cosine cutoff C must be between 0 and 1 (inclusive). If C = 0, no grazing approximation will be used; if C = 1, the grazing approximation will be used for every scoring particle. The substitute cosine S must be greater than or equal to 0. If S = 0, it will be reset to a huge number so that dividing by it makes the score essentially 0; in this way, particles in the grazing range can be ignored when diagnosing the tally. Normally, the ratio S/C is between one-half and two-thirds[2]; if it is outside 0.1% of those bounds, a warning is issued. If any cosine bins are within the grazing band, they may be incorrect[2]; a new warning is issued in this case.

The values of μ_{cut} and μ_{sub}, and the ratio μ_{sub}/μ_{cut}, are written in print table 30 and in the tally heading for each F2 tally.

III. Modifications

Nine source files were modified. A new global variable, `f2cut(2,ntal)`, was added to FDAC (fixed dynamically allocated common) part of module `mcnp_global`. `f2cut(1,i)` and `f2cut(2,i)` are the cosine cutoff and cosine substitute, respectively, for tally `i`. We were reluctant to define a variable that will likely be zero for most tallies in most problems, but there is precedent for this, `lft(mkft,ntal)`, also in FDAC, where `mkft` is the number of kinds of FT card special treatments. We note that the C and S of Sec. II are stored in and could be used from `tds`, but `f2cut` was added to handle the default values that will be used for F2 tallies without the MUC special treatment. Briefly, the files that were modified are:

mcnp_global.F90: add `f2cut` to FDAC.
setdas.F90: allocate `f2cut`; write, read, and broadcast `f2cut` with FDAC; deallocate `f2cut`.
mcnp_params.F90: change the number of kinds of FT card special treatments, `mkft`, from 16 to 17.
mcnp_data.F90: add 'muc' as the 17th entry in the names of FT-card special treatments array, `hft`.
nextit.F90: make default MUC values `-huge_float` in order to properly treat the jump character J.
itally.F90: add `half` as a used parameter; add a new entry, 2, to the `nz` array for the number of entries expected on special tally treatment cards; add default values for F2 tallies without the MUC special treatment; add error checking; add default values for the jump character J.
italpr.F90: print μ_{cut} and μ_{sub} for F2 tallies in print table 30.
tally.F90: remove `half` and `twenty` from the list of used parameters; use `f2cut` instead of `half` and `twenty` for type 2 tallies.
tallyq.F90: print μ_{cut} and μ_{sub} in the heading for F2 tallies.

The modified code passes the MCNP6 regression test set except for inp02, inp06, inp07, inp13, inp15, inp16, inp77, inp83, and inp85, all of which use F2 tallies; the only differences are that, in the modified code, μ_{cut} and μ_{sub} are written in the output files. The capability has been tested and works in continue runs.

A complete listing of the Unix diffs between the modified subroutines and the original, unmodified subroutines from the initial release of MCNP6 follows. These changes can easily be made in earlier versions of MCNP.

```
yr.master% diff -b mcnp_global.F90 /scratch3/fave/MCNP6_Initial_Release/Source/src/mcnp_global.F90
53d52
<      &  f2cut(:,:),    &  != cosine cutoff and substitute cosine for f2 tallies.
yr.master% diff -b setdas.F90 /scratch3/fave/MCNP6_Initial_Release/Source/src/setdas.F90
215c215
<      & egg, elp, esa, ewwg, excitation_mean_mev, f2cut, fdd, febl, fim, flc, flx, fma, &
---
>      & egg, elp, esa, ewwg, excitation_mean_mev, fdd, febl, fim, flc, flx, fma, &
413,415d412
<    allocate( f2cut( 1:2,  1:ntal1 ) ,stat=mstat)
<    if(mstat /=0) call get_mem_alloc_error('dyn_allocate','f2cut')
<    f2cut = zero
1282c1279
<      & eee, eek, egg, elp, esa, ewwg, excitation_mean_mev, f2cut, fim, flc, &
---
>      & eee, eek, egg, elp, esa, ewwg, excitation_mean_mev, fim, flc, &
1339c1336
<    write(iu)  esa,  ewwg, excitation_mean_MeV, f2cut, fim,  flc,  fmg
---
>    write(iu)  esa,  ewwg, excitation_mean_MeV, fim,  flc,  fmg
1474c1471
<      & eee, eek, egg, elp, esa, ewwg, excitation_mean_mev, f2cut, fim, flc, &
---
>      & eee, eek, egg, elp, esa, ewwg, excitation_mean_mev, fim, flc, &
1534c1531
<    read(iu)  esa,  ewwg, excitation_mean_MeV, f2cut, fim,  flc,  fmg
---
>    read(iu)  esa,  ewwg, excitation_mean_MeV, fim,  flc,  fmg
1873c1870
<      & excitation_mean_mev, f2cut, fim, flc, fma, fme, fmg, for, frc, fso, fst, ftt, gmg, &
---
>      & excitation_mean_mev, fim, flc, fma, fme, fmg, for, frc, fso, fst, ftt, gmg, &
1943d1939
<    if( allocated( f2cut ) )  call msg_bcast( mf,  f2cut )
2409c2405
<      & eek, egg, elp, esa, ewwg, excitation_mean_mev, f2cut, fdd, febl, fim, &
---
>      & eek, egg, elp, esa, ewwg, excitation_mean_mev, fdd, febl, fim, &
2550,2552d2545
<    if (  allocated   ( f2cut   ) ) then
<      deallocate (  f2cut )
<    endif
yr.master% diff -b mcnp_params.F90 /scratch3/fave/MCNP6_Initial_Release/Source/src/mcnp_params.F90
79c79
<    integer, parameter, public :: mkft   = 17      != Number of kinds of FT card special treatments.
---
>    integer, parameter, public :: mkft   = 16      != Number of kinds of FT card special treatments.
yr.master% diff -b mcnp_data.F90 /scratch3/fave/MCNP6_Initial_Release/Source/src/mcnp_data.F90
132c132
<      &    'phl', 'cap', 'res', 'tag', 'let', 'pds','roc', 'muc' /)
---
>      &    'phl', 'cap', 'res', 'tag', 'let', 'pds','roc' /)
yr.master% diff -b nextit.F90 /scratch3/fave/MCNP6_Initial_Release/Source/src/nextit.F90
991,993d990
<           else if( hitm=='muc' ) then
<              rtp(ipl+nwc+1) = -huge_float
<              rtp(ipl+nwc+2) = -huge_float
yr.master% diff -b itally.F90 /scratch3/fave/MCNP6_Initial_Release/Source/src/itally.F90
27c27
<      & jtf, jun, ksd, ksm, lat, lca, lft, lja, mfl, ncl, nmt, f2cut, rtp, npq,  &
---
>      & jtf, jun, ksd, ksm, lat, lca, lft, lja, mfl, ncl, nmt, rtp, npq,         &
38c38
<      & I_TAL_FLAG1, HALF
---
>      & I_TAL_FLAG1
70d69
<    real(dknd), parameter :: dc = tenth  ! default value of mu_cut (used twice)
89c88
<    integer :: nz(mkft) = (/ 3,3,2,0,0,1,0,1,0,-1,-1,-1,1,-1,1,-1,2 /)
---
>    integer :: nz(mkft) = (/ 3,3,2,0,0,1,0,1,0,-1,-1,-1,1,-1,1,-1 /)
122,126d120
<      ! set mu_cut and mu_sub defaults.
```

```
<     if( iy == 2 )then
<       f2cut(1,ital) = dc
<       f2cut(2,ital) = half*f2cut(1,ital)
<     endif
1644,1651d1637
<       ! FT MUC option: mu_cut and mu_sub for f2 tallies.
<       ! note: these are duplicated in tds(lft(17,ital)+1) and tds(lft(17,ital)+2),
<       ! but not if there is no ft muc option.
<       if( k == 17 ) then
<         f2cut(1,ital) = rtp(l+i+1)
<         f2cut(2,ital) = rtp(l+i+2)
<       end if
<
1743,1779d1728
<       ! FT MUC option:
<       if( lft(17,ital) /= 0 ) then
<          if( jptal(2,ital) /= 2 ) then
<            call erprnt(1,1,1,kl,0,0,0,0,&
<      &         '  "ft muc can be used only with f2 tallies; tally",i9,"."')
<            go to 700
<          end if
<          if( f2cut(1,ital) == -huge_float ) then
<            f2cut(1,ital) = dc
<          end if
<          if( f2cut(2,ital) == -huge_float ) then
<            f2cut(2,ital) = half*f2cut(1,ital)
<          end if
<          if( f2cut(1,ital) < zero .or. f2cut(1,ital) > one )then
<            call erprnt(1,1,1,kl,0,0,0,0,&
<      &         '  "cosine cutoff must be >= 0 and <= 1 for ft muc, tally",i9,"."')
<          end if
<          if( f2cut(2,ital) < zero )then
<            call erprnt(1,1,1,kl,0,0,0,0,&
<      &         '  "cosine substitute must be >= 0 for ft muc, tally",i9,"."')
<          else if( f2cut(2,ital) == zero )then
<       ! if mu_sub is input as 0., reset to huge.
<            call erprnt(1,2,1,kl,0,0,0,0,&
<      &         '  "cosine substitute set to huge for ft muc, tally",i9,"."')
<            f2cut(2,ital) = huge_float
<          end if
<          if( f2cut(1,ital) > zero )then
<            if( f2cut(2,ital)/f2cut(1,ital) < 0.4995 .or. &
<      &      (f2cut(2,ital)/f2cut(1,ital) > 0.667333 .and. &
<      &      f2cut(2,ital) /= huge_float) )then
<             call erprnt(1,2,1,kl,0,0,0,0,&
<      &          '  "normally substitute/cutoff is between 1/2 and 2/3; ft muc, tally", &
<      &          i9,  "."')
<            end if
<          end if
<       end if
<
2003,2017d1951
<     ! warn if any F2 cosine bins are within the grazing range.
<     if( jptal(2,ital) == 2 )then
<       n = iptal(6,4,ital)-1
<       if( n > 1)then
<         l = iptal(6,1,ital)
<         do i=2,n
<           if( abs(tds(l+i-1)) > zero .and. abs(tds(l+i-1)) < f2cut(1,ital)) then
<             call erprnt(1,2,1,kl,0,0,0,0, &
<      &          '"surface fluxes within grazing range may be incorrect; tally",i9,"."')
<             exit
<           end if
<         end do
<       end if
<     end if
<
yr.master% diff -b italpr.F90 /scratch3/fave/MCNP6_Initial_Release/Source/src/italpr.F90
10c10
<     & nmt, f2cut, rtp, tds
---
>     & nmt, rtp, tds
245,255d244
<   ! print mu_cut and mu_sub for f2 tallies.
```

```
<   if( jptal(2,ital)==2 )then
<     if( f2cut(1,ital) == zero )then
<       write(iuo,'(/,1x,"cosine cutoff =",1pe12.5,2x,"substitute =",e12.5, &
<      &   2x,"sub/cut = n/a")') f2cut(1,ital),f2cut(2,ital)
<      else
<       write(iuo,'(/,1x,"cosine cutoff =",1pe12.5,2x,"substitute =",e12.5, &
<      &   2x,"sub/cut =",e12.5)') f2cut(1,ital),f2cut(2,ital),f2cut(2,ital)/f2cut(1,ital)
<      end if
<   endif
<
yr.master% diff -b tally.F90 /scratch3/fave/MCNP6_Initial_Release/Source/src/tally.F90
13c13
<     & lca, lft, lja, lme, mat, mfl, ncl, ntbb, nxs, f2cut, rtc, tal, tds, &
---
>     & lca, lft, lja, lme, mat, mfl, ncl, ntbb, nxs, rtc, tal, tds, &
16,17c16,17
<   use mcnp_params, only : dknd, half, huge_float, one, thousandth, &
<     & two, vsmall, zero
---
>   use mcnp_params, only : dknd, half, huge_float, one, tenth, thousandth, &
>     & twenty, two, vsmall, zero
280c280
<       if( abs(dr)>=f2cut(1,ital) ) then
---
>       if( abs(dr)>=tenth ) then
283c283
<         ta = ta/f2cut(2,ital)
---
>         ta = twenty*ta
yr.master% diff -b tallyq.F90 /scratch3/fave/MCNP6_Initial_Release/Source/src/tallyq.F90
9c9
<   use mcnp_global, only : iptal, itds, jptal, lft, f2cut, tds
---
>   use mcnp_global, only : iptal, itds, jptal, lft, tds
163,173d162
<   ! print mu_cut and mu_sub for f2 tallies.
<   if( jptal(2,ital)==2 )then
<     if( f2cut(1,ital) == zero )then
<       write(iuo,'(/,11x,"cosine cutoff =",1pe12.5,2x,"substitute =",e12.5, &
<      &   2x,"sub/cut = n/a")') f2cut(1,ital),f2cut(2,ital)
<      else
<       write(iuo,'(/,11x,"cosine cutoff =",1pe12.5,2x,"substitute =",e12.5, &
<      &   2x,"sub/cut =",e12.5)') f2cut(1,ital),f2cut(2,ital),f2cut(2,ital)/f2cut(1,ital)
<      end if
<   endif
<
```

IV. Example Input File

As an example and for further documentation, this is the input file that was used for the test problem of Ref. 2.

```
graphite-reflected u235 sphere
10   92  -18.74  -10         imp:n=1
11    6  -2.1    -11 10      imp:n=1
99    0           11         imp:n=0

10   so    6.
11   so   16.

mode n
mgopt f 30
rand seed=100000000101
kcode 160000  1.  20 1620
totnu no  $ to compare with partisn
sdef pos 0. 0. 0.  rad=d1 erg=d2
si1  0.  6.
sp1  -21 2
sp2  -3
m92   92235.50m  1.
m6     6012.50m  1.
c000  0.  2.44437E-02 4.88729E-02 7.32728E-02 9.76289E-02 1.21927E-01
          1.46152E-01 1.00000E+00 t
```

```
 f001:n  11
tf001 j j j j j 2 j j
c these bins match a one-d s64 quadrature set
 c001 0.  1.e-12
         2.44437E-02 4.88729E-02 7.32728E-02 9.76289E-02 1.21927E-01
         1.46152E-01 1.70289E-01 1.94325E-01 2.18245E-01 2.42034E-01
         2.65678E-01 2.89164E-01 3.12477E-01 3.35603E-01 3.58529E-01
         3.81240E-01 4.03723E-01 4.25965E-01 4.47953E-01 4.69673E-01
         4.91112E-01 5.12257E-01 5.33097E-01 5.53617E-01 5.73807E-01
         5.93654E-01 6.13146E-01 6.32272E-01 6.51020E-01 6.69378E-01
         6.87337E-01 7.04885E-01 7.22011E-01 7.38706E-01 7.54959E-01
         7.70761E-01 7.86103E-01 8.00975E-01 8.15367E-01 8.29273E-01
         8.42683E-01 8.55589E-01 8.67984E-01 8.79860E-01 8.91211E-01
         9.02028E-01 9.12307E-01 9.22040E-01 9.31222E-01 9.39847E-01
         9.47911E-01 9.55408E-01 9.62335E-01 9.68686E-01 9.74458E-01
         9.79647E-01 9.84251E-01 9.88267E-01 9.91692E-01 9.94525E-01
         9.96763E-01 9.98405E-01 9.99451E-01 1.00000E+00 t
c musub/mucut = 1/2
 f002:n  11
tf002 j j j j j 2 j j
 c002 0.  2.44437E-02 4.88729E-02 7.32728E-02 9.76289E-02 1.21927E-01
         1.46152E-01 1.70289E-01 1.94325E-01 2.18245E-01 2.42034E-01
         2.65678E-01 2.89164E-01 3.12477E-01 3.35603E-01 3.58529E-01
         3.81240E-01 4.03723E-01 4.25965E-01 4.47953E-01 4.69673E-01
         4.91112E-01 5.12257E-01 5.33097E-01 5.53617E-01 5.73807E-01
         5.93654E-01 6.13146E-01 6.32272E-01 6.51020E-01 6.69378E-01
         6.87337E-01 7.04885E-01 7.22011E-01 7.38706E-01 7.54959E-01
         7.70761E-01 7.86103E-01 8.00975E-01 8.15367E-01 8.29273E-01
         8.42683E-01 8.55589E-01 8.67984E-01 8.79860E-01 8.91211E-01
         9.02028E-01 9.12307E-01 9.22040E-01 9.31222E-01 9.39847E-01
         9.47911E-01 9.55408E-01 9.62335E-01 9.68686E-01 9.74458E-01
         9.79647E-01 9.84251E-01 9.88267E-01 9.91692E-01 9.94525E-01
         9.96763E-01 9.98405E-01 9.99451E-01 1.00000E+00 t
 f012:n  11
tf012 j j j j j 2 j j
ft012  muc  0.01          j
 f022:n  11
tf022 j j j j j 2 j j
ft022  muc  9.76289E-02   4.88145E-02
 f032:n  11
tf032 j j j j j 2 j j
ft032  muc  7.32728E-02   3.66364E-02
f042:n  11
tf042 j j j j j 2 j j
ft042  muc  4.88729E-02   2.44364E-02
 f052:n  11
tf052 j j j j j 2 j j
ft052  muc  2.44437E-02   1.22219E-02
 f062:n  11
tf062 j j j j j 2 j j
ft062  muc  1.e-12        5.e-13
 f072:n  11
c musub/mucut = 2/3
tf072 j j j j j 2 j j
ft072  muc  0.1           0.0666666666666
 f082:n  11
tf082 j j j j j 2 j j
ft082  muc  0.01          0.00666666666666
 f092:n  11
tf092 j j j j j 2 j j
ft092  muc  9.76289E-02   6.50859E-02
 f102:n  11
tf102 j j j j j 2 j j
ft102  muc  7.32728E-02   4.88485E-02
 f112:n  11
tf112 j j j j j 2 j j
ft112  muc  4.88729E-02   3.25819E-02
 f122:n  11
tf122 j j j j j 2 j j
ft122  muc  2.44437E-02   1.62958E-02
 f132:n  11
tf132 j j j j j 2 j j
ft132  muc  1.e-12        6.66667E-13
c musub/mucut from quadratic fit
```

```
f142:n  11
ft142  muc  1.e-12        5.e-13
 f152:n  11
ft152  muc  0.01          6.498567E-03
 f162:n  11
ft162  muc  2.44437E-02   1.601688E-02
 f172:n  11
ft172  muc  4.88729E-02   3.190795E-02
 f182:n  11
ft182  muc  7.32728E-02   4.746107E-02
 f192:n  11
ft192  muc  9.76289E-02   6.259907E-02
 f202:n  11
ft202  muc  0.1           6.404901E-02
c
print

end of input
```

V. Summary and Conclusions

The new MUC special tally treatment for surface flux tallies will allow users to diagnose and overcome difficulties associated with grazing. One immediate benefit for any problem is that users can turn off the grazing approximation altogether by setting the first entry (C) to 0. If the variance and the variance of the variance behave reasonably, then the tally is well behaved and doesn't need a grazing approximation.

A method for diagnosing surface flux tallies is suggested in Ref. 2. It consists of calculating the quantity of interest as a function of μ_{cut} (C in Sec. II) for a fixed μ_{sub} (S in Sec. II). When μ_{cut} can be decreased without affecting the tally, that indicates that the tally is accurate. It is easier to find the appropriate value of μ_{cut} when an accurate μ_{sub} is used. A value of $\mu_{sub} = \mu_{cut}/2$ is often the most accurate one, but, as shown in Ref. 2, sometimes a more accurate one is $\mu_{sub} = 2\mu_{cut}/3$. The most accurate one depends on the value of μ_{cut} and the true angular flux distribution (in angle) and may be neither $\mu_{cut}/2$ nor $2\mu_{cut}/3$.

The MUC capability has been integrated with the LINES capability[4] for surface and volume integrals of uncollided adjoint fluxes and forward-adjoint flux products in the MCNP6 initial release[3] (MCNP6_Beta1).

In general, hard-wired values of μ_{cut} and/or μ_{sub} will not allow MCNP to compute analytic benchmark values for surface flux tallies. The simple modifications documented here will allow the code to handle such problems, with proper care by the user.

References

1. Jeffrey A. Favorite, Ashley D. Thomas, and Thomas E. Booth, "On the Accuracy of a Common Monte Carlo Surface Flux Grazing Approximation," *Nuclear Science and Engineering*, **168**, *2*, to be published (2011).

2. Jeffrey A. Favorite, "Monte Carlo Surface Flux Tallies," *Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, Rio de Janeiro, Brazil, May 8-12, CD-ROM (2011).

3. Tim Goorley, Mike James, Tom Booth, Forrest Brown, Jeff Bull, Larry Cox, Joe Durkee, Jay Elson, Mike Fensin, Art Forster, John Hendricks, Grady Hughes, Russ Johns, Brian Kiedrowski, Roger Martz, Stepan Mashnik, Gregg McKinney, Denise Pelowitz, Dick Prael, Jeremy Sweezy, Laurie Waters, Trevor Wilcox, and Tony Zukaitis, "MCNP6 Initial Release Notes: MCNP6_Beta1," LA–UR–11–1765, Los Alamos National Laboratory (March 2011).

4. Jeffrey A. Favorite, "Surface and Volume Integrals of Uncollided Adjoint Fluxes and Forward-Adjoint Flux Products in Arbitrary Three-Dimensional Geometries Using MCNP," *Transactions of the American Nuclear Society*, **101**, 633-635 (2009).

JAF:jaf

Distribution:
R. C. Little, XCP-DO, MS B259, `rcl@lanl.gov`
A. Sood, XCP-7, MS A143, `sooda@lanl.gov`
D. A. Pimentel, HPC-1, MS F663, `davidp@lanl.gov`
M. L. Fensin, D-5, MS K575, `mfensin@lanl.gov`
J. S. Hendricks, D-5, MS K575, `jxh@lanl.gov`
M. R. James, D-5, MS K575, `mrjames@lanl.gov`
G. W. McKinney, D-5, MS K575, `gwm@lanl.gov`
S. A. Turner, CCS-2, MS D409, `turners@lanl.gov`
S. C. Frankle, XTD-4, MS T082, `frankles@lanl.gov`
F. B. Brown, XCP-3, MS A143, `fbrown@lanl.gov`
J. S. Bull, XCP-3, MS A143, `jsbull@lanl.gov`
R. A. Forster, XCP-3, MS A143, `raf@lanl.gov`
J. T. Goorley, XCP-3, MS A143, `jgoorley@lanl.gov`
H. G. Hughes, XCP-3, MS A143, `hgh@lanl.gov`
B. C. Kiedrowski, XCP-3, MS A143, `bckiedro@lanl.gov`
D. K. Parsons, XCP-5, MS F663, `dkp@lanl.gov`
G. P. Estes, XCP-7, MS F663, `gpe@lanl.gov`
J. P. Lestone, XCP-7, MS F663, `lestone@lanl.gov`
E. F. Shores, XCP-7, MS F663, `eshores@lanl.gov`
C. J. Solomon, XCP-7, MS F663, `csolomon@lanl.gov`
B. A. Temple, XCP-7, MS F663, `temple@lanl.gov`
J. A. Favorite, XCP-7, MS F663, `fave@lanl.gov`
X-Archive, `xarchive@lanl.gov`
XCP-DO File
XCP-7 File